

RANCANG BANGUN APLIKASI MANAJEMEN PROYEK UNTUK PEMANTAUAN DAN MENGELOLA BATAS WAKTU PADA BLUNG STUDIO

Dhoddy Cahya Putranto

**Sistem Informasi, Ilmu Komputer, Universitas Dian Nuswantoro, Jl. Nakula No. 5-11,
Semarang, 50181**

dodi.cahya@gmail.com

ABSTRAK: *Perkembangan yang cukup pesat ini menyebabkan pihak manajemen harus mampu mengelola sumber daya yang dimiliki untuk menghasilkan produk yang berkualitas tinggi serta mampu bersaing di pasar. Untuk itu peranan manajemen proyek atas TI sangat dibutuhkan untuk memperbaiki kinerja TI yang ada di perusahaan agar dapat mendukung strategi bisnis.*

Membangun aplikasi project management pada Blung Studio untuk mengelola proyek perangkat lunak agar pekerjaan dapat dikelola dan pekerjaan selesai tepat pada waktunya. Setiap proyek yang dikerjakan dapat dikelola dengan baik, developer mengetahui apa yang harus dikerjakan, owner atau project manager mengetahui perkembangan proyek yang sedang dikerjakan sampai menjadi sebuah produk.

Dengan menerapkan metode scrum dirasa lebih cocok karena pengerjaan proyek dibagi dalam bentuk tim. Proyek atau pekerjaan dibagi menjadi beberapa bagian yang dibagi kepada anggota tim yang mampu mengerjakan sehingga pekerjaan dirasa lebih mudah dikerjakan dan mampu diselesaikan dengan cepat. Diharapkan dengan menggunakan aplikasi project management ini mampu membantu kinerja para developer.

KATA KUNCI : *Scrum, TI, Manajemen Proyek*

ABSTRACT : A fairly rapid growth is causing management to be able to manage its resources to produce a high quality product and be able to compete in the market. For that role over the IT project management is needed to improve the performance of existing IT in order to support the company's business strategy.

Building a project management application on Blung Studio to manage software projects so that the work can be managed and the work completed on time. Each project done properly managed, developers know what to do, the owner or project manager to know the progress of projects being worked on until it becomes a product.

By applying Scrum method is considered more suitable for project work is divided in the form of the team. Project or the work is divided into sections and divided by team who are able to do so the work feels easier to do and can be completed quickly. It is expected to use project management application is able to assist the performance of the developers

KEYWORD : Scrum, IT, Project Management

I. PENDAHULUAN

Perkembangan bisnis dari organisasi dengan tingkat kompetisi yang tinggi telah mending internal manajemen perusahaan untuk melakukan evaluasi terhadap pemanfaatan teknologi informasi. Perbaikan secara terus menerus harus dilakukan dengan melihat peluang-peluang dari setiap bagian didalam organisasi yang membutuhkan peningkatan kerja. Setiap pengeluaran investasi atau biaya atas pengadaan teknologi informasi harus mempertimbangkan sisi keuntungan yang akan diperoleh. Untuk itu peranan manajemen proyek atas TI sangat dibutuhkan untuk memperbaiki kinerja TI yang ada di perusahaan agar dapat mendukung strategi bisnis dan operasional perusahaan.[1]

Perkembangan yang cukup pesat ini menyebabkan pihak manajemen harus mampu mengelola sumber daya yang dimiliki untuk menghasilkan produk yang berkualitas tinggi serta mampu bersaing di pasar. Kemampuan pihak manajemen untuk menghasilkan produk yang berkualitas dengan adanya keterbatasan terhadap waktu, biaya dan ruang lingkup pekerjaan harus didukung oleh pemahaman mengenai manajemen proyek yang baik. Sebuah aplikasi Project Management untuk saat ini sangat diperlukan oleh sebuah perusahaan atau badan usaha. Tidak dapat dipungkiri dengan semakin banyaknya proyek atau pekerjaan yang datang tentunya kita dibuat ribet dalam memprioritaskan proyek mana dulu yang akan dikerjakan, proyek mana yang bisa didelegasikan kepada orang lain ataupun tanggal deadline project yang ternyata kita sendiri lupa. Project Management sebagai suatu sistem yang dapat membantu dalam pengorganisasian suatu pekerjaan merupakan hal positif bagi badan usaha yang bergerak di bidang teknologi informasi.

Proyek perangkat lunak yang ditangani BlungStudio dirasa masih dikelola kurang baik, yaitu developer atau karyawan hanya mendapatkan list proyek perangkat lunak apa saja yang akan dikerjakan tanpa melihat prioritas proyek mana yang harus dikerjakan

dulu berdasarkan deadline project. Dan juga tidak adanya penjadwalan sebagai penunjang

Dengan adanya aplikasi project management ini diharapkan dapat mengatasi masalah yang ada. Aplikasi project management ini juga terdapat deadline project, sehingga diharapkan proyek yang dikerjakan dapat diselesaikan tepat pada waktunya. Dengan demikian penulis memiliki ide untuk membuat aplikasi ini dengan judul **“Rancang Bangun Aplikasi Project Management pada Blung Studio”**

II. LANDASAN TEORI

2.1 Manajemen Proyek / Project Management

Pengertian manajemen proyek menurut Budi santoso (2003;3) manajemen proyek adalah kegiatan merencanakan, mengorganisasikan, mengarahkan dan mengendalikan sumber daya organisasi perusahaan untuk mencapai tujuan tertentu dalam waktu tertentu dengan sumber daya tertentu. Manajemen proyek adalah aplikasi pengetahuan (knowledges), Keterampilan (skills), alat (tools) dan teknik (techniques) dalam aktifitas – aktifitas proyek untuk memenuhi kebutuhan – kebutuhan proyek (PMBOK, 2004) [2].

2.1.1 Manajemen Proyek Perangkat Lunak
Manajemen proyek perangkat lunak merupakan bagian yang penting dalam pembangunan perangkat lunak. Sekalipun tidak bersifat teknis seperti pengkodean, hal-hal dalam manajemen proyek PL ini mampu menentukan apakah proyek akan berjalan dengan baik sehingga menghasilkan produk yang baik. Hal-hal yang berkaitan dengan manajemen adalah pengelolaan personel dan koordinasi tim, proses, pengukuran proyek-termasuk menentukan harga dari PL, penjadwalan dan sebagainya.

2.2 Web Server

Server web atau peladen web dapat merujuk baik pada perangkat keras ataupun perangkat lunak yang menyediakan layanan

akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs web dalam layanan ke pengguna dengan menggunakan aplikasi tertentu seperti peramban web.

Penggunaan paling umum server web adalah untuk menempatkan situs web, namun pada prakteknya penggunaannya diperluas sebagai tempat penyimpanan data ataupun untuk menjalankan sejumlah aplikasi kelas bisnis [3].

2.3 PHP

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

2.4 MySQL

MySQL (My Structured Query Language) atau yang biasa dibaca **mai-se-kuel** adalah sebuah program pembuat dan pengelola database atau yang sering disebut dengan DBMS (*DataBase Management System*), sifat dari DBMS ini adalah *Open Source*, anda bias mendapatkannya secara gratis pada alamat <http://www.mysql.com>. Saat ini mendengar open source maka seakan kita ingat dengan sistem operasi handal keturunan Unix yaitu Linux.

MySQL sebenarnya produk yang berjalan pada platform Linux, dengan adanya perkembangan dan banyaknya pengguna, serta lisensi dari database ini alaaah Open Source, maka para pengembang kemudian merilis versi Windows. Selain itu MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi Multi User (Banyak Pengguna). Kelebihan lain dari MySQL adalah menggunakan bahasa query (permintaan) standar SQL (*Structured Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur, SQL telah distandarkan untuk semua program pengakses database seperti oracle, PosgreSQL, SQL Server dan lain-lain. Sebagai sebuah program penghasil database, MySQL tidak mungkin berjalan sendiri tanpa adanya sebuah aplikasi pengakses database yang dihasilkan. MySQL dapat didukung oleh hampir semua program aplikasi baik yang Open Source seperti PHP maupun tidak Open Source yang ada pada platform windows seperti Visual Basic, Delphi, dan lain-lain [4].

- Beberapa keunggulan dari MySQL adalah:
 - a. Mampu menangani jutaan user dalam waktu yang bersamaan.
 - b. Mampu menampung lebih dari 50.000.000 record.
 - c. Sangat cepat mengeksekusi perintah.
 - d. Memiliki user privilege yang mudah dan efisien.

2.5 UML

Unified modeling language (UML) adalah sebuah bahasa yang telah menjadi *standart* dalam *industry* visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML dapat digunakan dalam pembuatan model untuk semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta di tulis dalam bahasa pemrograman apapun. Model yang dimaksud adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Tetapi karena UML mempunyai

konsep dasar *class* dan *operation*, maka UML lebih cocok digunakan untuk pemrograman *software* yang menggunakan bahasa-bahasa pemrograman yang berorientasi objek, seperti C++, java, C#, atau VB.NET.

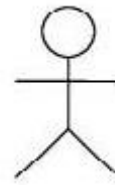
Unified Modelling Language (UML) meliputi simbol-simbol dan suatu tata bahasa yang menjelaskan bagaimana simbol-simbol tersebut dapat digunakan. Dengan mempelajari simbol-simbol dan tata bahasa, maka tiap orang dapat memahami suatu gambaran dokumentasi model yang sebelumnya hanya orang-orang tertentu saja yang memahaminya. UML merupakan bahasa pemodelan yang *universal* yang digunakan sebagai sarana komunikasi untuk bertukar gagasan dalam perancangan suatu *software*. [5]

2.5.1 Komponen *Unified Modelling Language* (UML)

Unified modeling language (UML) terbagi menjadi sembilan jenis diagram yang masing-masing memiliki aturan-aturan tertentu dalam penyusunannya. Diagram-diagram tersebut tersusun atas sejumlah elemen grafis yang saling membentuk satu kesatuan dalam pemodelan *software*. Masing-masing diagram UML mempresentasikan berbagai sudut pandang terhadap sistem dan mendefinisikan apa yang dikerjakan oleh sistem bukan bagaimana cara sistem bekerja.

Unified Modelling Language (UML) memiliki banyak diagram agar dapat memodelkan sistem secara lebih akurat, karena dalam pembuatan suatu sistem terdapat sejumlah pihak yang berkepentingan (*stakeholder*) pada aspek-aspek yang berlainan dari sistem. Oleh karena itu dengan adanya diagram-diagram pada UML yang mempunyai fungsi masing-masing sesuai bidang dari pihak-pihak yang terlibat dalam pembuatan sistem, maka pihak-pihak yang terlibat dapat memahami informasi yang ingin disampaikan dari satu pihak menuju pihak lainnya. Meskipun UML

mempunyai bai semua diagram pembuatan s pengerjaan Ti



etapi tidak kan dalam Dalam ini hanya digunakan 4 (dua) diagram yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* untuk mendukung desain dari sistem. Berikut penjelasannya:

1. Use case diagram

Merupakan diagram yang dibuat pada awal pemodelan suatu *software*, karena diagram ini memberikan penjelasan (*high level*) antara sistem dengan “dunia luar” serta fitur-fitur apa yang harus dimiliki oleh sistem dipandang dari dunia luar tersebut. Selain itu *use case* memiliki suatu bentuk naratif yang memberikan *standard* informasi yang diperlukan dalam analisis, desain dan *coding* fitur-fitur tersebut. Suatu *use case* diagram komponen berupa:

a. Stereotype

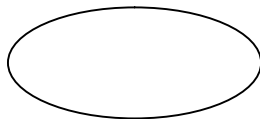
Stereotype menyediakan informasi tentang peranan dari elemen tanpa menyebutkan imlementasinya. Digunakan untuk menggambarkan *use case dependency*, *class*, *package* dan *classifier*. *Stereotype* merupakan notasi yang dalam diagram dituliskan dengan *Guillemet* “<<...>>”

b. Actor

Actor mewakili pihak yang terlibat dalam suatu *use case* yang terjadi. *Actor* dapat berupa manusia, sistem, atau *device* yang memiliki peranan dalam keberhasilan operasi dari sistem. Simbol untuk *actor* manusia dapat dilihat pada gambar 2.1

c. Use Case

Use Case mengidentifikasi fitur utama (kunci) dari sistem. Tanpa *use case*, sistem tidak dapat memenuhi permintaan *user/actor*. Tiap *use case* mendefinisikan dan menggambarkan *goal* yang harus di capai oleh sistem. Suatu *use case* difokuskan pada *goal* yang ingin dicapai, bukan pada bagaimana cara mengimplementasikan *goal* tersebut. Simbol *use case*:



d. Connector

1. Asosiasi, mengidentifikasi interaksi antara setiap *actor* tertentu dengan setiap *use case* tertentu. Tiap asosiasi menjadi dialog yang harus dijelaskan dalam *use case narrative*. Asosiasi bisa berarah (garis dengan anak panah) jika komunikasi satu arah dan dapat juga tidak berarah jika komunikasi terjadi ke dua arah.
2. Generalisasi, untuk mendefinisikan relasi antara dua *actor* atau dua *use case* yang mana salah satunya menurunkan dan menambahkan sifat dari yang lainnya. Digambarkan menggunakan garis bermata panah kosong dari yang menurunkan mengarah ke yang diturunkan.
3. Dependensi, terdiri dari dua jenis yaitu :
 - a. Include, mengidentifikasi hubungan antar dua *use case* di mana



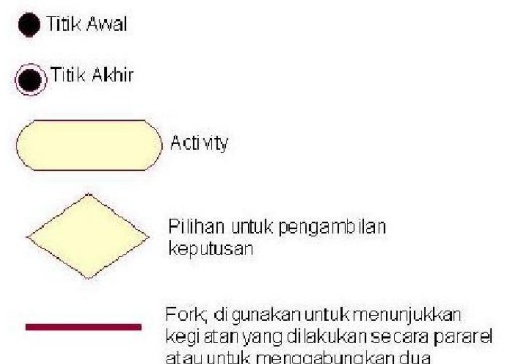
gil
in.
an
.us
ah

notasi
<<include>> pada
garis. Arah mata
panah sesuai dengan
arah pemanggilan
(umumnya mengarah
ke kanan).

- b. Extend, terjadi jika pemanggilan memerlukan adanya kondisi tertentu. Digambarkan serupa dengan dependensi <<include>> kecuali arah panah berlawanan dan kata "include" diganti dengan "extend".

2. Activity diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Diagram ini bersifat dinamis, Diagram ini adalah tipe khusus dari *state diagram* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan member tekanan pada aliran kendali antar objek.



3. Sequence diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang di hasilkan. Masing-masing objek, termasuk *actor*, memiliki *lifeline vertical*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* di petakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. [6]

4. Class Diagram

Class adalah sebuah spesifikasi yang jika di instansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan,

asosiasi, dan lain lain. *Class* memiliki tiga area pokok:

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, hanya dapat dipanggil dari luar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. Public, dapat dipanggil oleh siapa saja



Hubungan antar class :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..")
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru,

- sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang telah dijelaskan di atas.

III. METODELOGI PENELITIAN

3.1 Dasar Penelitian

Penelitian ini dilakukan berdasarkan permasalahan yang telah di uraikan pada bab sebelumnya. Metode yang digunakan oleh penulis untuk menyusun penelitian ini adalah Agile Method: *Scrum*.

3.2 Metode Pengumpulan Data

Untuk membangun aplikasi project management, beberapa teknik pengumpulan data dan variable digunakan, diantaranya adalah:

1. Teknik Observasi
Teknik observasi adalah teknik pengamatan dan peninjauan secara langsung pada obyek penelitian yaitu pada Blung Studio.
2. Studi Pustaka
Cara pengumpulan data dengan mengumpulkan dan mempelajari dokumen-dokumen dan media referensi lainnya yang berhubungan dengan project management untuk mendapatkan teori yang dibutuhkan.

3.3 Sumber Data

Sumber data yang digunakan untuk membangun dan menguji coba sistem ini yang diperoleh dari hasil survey lapangan.

3.4 Jenis dan Sumber Data

Jenis dan sumber data yang menjadi bahan dalam kegiatan ini terdiri dari :

- a. **Data Primer** data dan informasi yang diperoleh berupa hasil observasi yang diinput langsung

menggunakan aplikasi yang telah dibangun.

- b. **Data Sekunder**, data dan informasi yang diperoleh dari dokumen, publikasi, laporan penelitian dari instansi manapun sumber data lainnya yang menunjang.

3.5 Ruang Lingkup Penelitian

Dalam melaksanakan penelitian ini, tempat yang dijadikan bahan penelitian yaitu Blung Studio. Waktu penelitian dilakukan pada bulan Maret 2013.

3.6 Model Perancangan Sistem

Untuk Pemodelan perancangan menggunakan UML diagram.

1. Use case diagram

Diagram ini memberikan gambaran dari fungsi sistem secara keseluruhan jika dipandang dari luar sistem dengan memberikan gambaran tentang aktor yang berperan dan use case yang tampak serta hubungan yang terjadi. Aktor yang ditentukan dalam tahapan ini adalah Owner, Project Manager, dan Developer.

2. Activity diagram

Activity Diagram ini memperlihatkan aksi yang terjadi pada saat user menyelesaikan suatu inputan untuk diberikan ke user yang lain. Semua aksi yang akan dilakukan oleh user dapat digambarkan melalui activity diagram ini. Mulai dari input project sampai dengan project itu selesai dikerjakan.

3. Sequence diagram

Diagram ini merupakan diagram yang digunakan untuk menunjukkan interaksi yang terjadi antar objek dalam suatu urutan waktu. Interaksi antar objek disini dapat diartikan sebagai pemanggilan suatu metode atau operasi pada obyek itu sendiri atau pada objek lain.

4. Class Diagram

Diagram ini memberikan penjelasan secara rinci mengenai deskripsi tiap kelas dan juga hubungan antar kelas didalam sebuah sistem. Suatu kelas diagram kelas memiliki tiga bagian pokok, yaitu nama, atribut, dan metode atau operasi yang dapat dilakukan kelas tersebut.

3.7 Metode Pengembangan Perangkat Lunak

3.7.1 SCRUM

Scrum: Sebuah kerangka kerja dimana pihak-pihak dapat mencari jalan keluar dari permasalahan yang kompleks dan pada saat yang bersamaan membuat produk yang memiliki nilai setinggi mungkin secara produktif dan kreatif. Scrum bersifat:

1. Sederhana
2. Mudah untuk dimengerti
3. Susah untuk dikuasai

Scrum adalah sebuah proses kerangka kerja yang telah digunakan semenjak tahun 1990 untuk mengelola pengembangan produk yang kompleks. Scrum bukanlah sebuah proses atau teknik untuk membuat produk, melainkan sebuah kerangka kerja dimana didalamnya kita dapat memasukkan berbagai proses dan teknik. Scrum akan menunjukkan hasil dari praktik pengelolaan dan pengembangan produk sehingga kita dapat terus menjadi lebih baik. [7](Ken Schwaber and Jeff Sutherland, 2013) Ada 3 elemen organisasi utama pada scrum yaitu *Product Owner*, *Scrum Master*, dan *the Scrum team*. *Scrum Master* dapat dianggap sebagai pelatih bagi tim, membantu anggota tim menggunakan kerangka Scrum untuk tampil di tingkat tertinggi. *Product Owner* mewakili bisnis, pelanggan atau pengguna dan memandu tim ke arah pengembangan produk yang tepat. Sedangkan *The Scrum Team* merupakan grup pengembang kecil biasanya terdiri dari 5-9 orang. Untuk proyek yang sangat

besar, pekerjaan biasanya dibagi dan didelegasikan ke grup-grup kecil. Jika sangat dibutuhkan Scrum Master juga dapat ikut membantu dalam koordinasi team.[8](Henrik Kniberg, Ifnu Bima, 2007)

Tim Scrum (*The Scrum Team*)

Tim Pengembang terdiri dari para ahli yang bekerja untuk menghasilkan potongan produk “Selesai” yang berpotensi untuk dirilis di setiap akhir Sprint. Hanya anggota tim yang menyelesaikan potongan produk ini.

Tim Pengembang distrukturisasi dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Sinergi yang dihasilkan akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan. Tim Pengembang memiliki karakteristik sebagai berikut:

- Mereka mengatur dirinya sendiri. Tidak ada satu orang pun (bahkan Scrum Master sekalipun) yang memerintah Tim Pengembang bagaimana caranya menghasilkan potongan fungsionalitas dari Product Backlog yang berpotensi untuk dirilis.
- Tim Pengembang bersifat antar lintas, yang artinya semua keahlian yang dibutuhkan untuk menghasilkan potongan dari produk harus ada di dalam tim.
- Scrum tidak mengenal adanya jabatan tertentu untuk anggota tim selain Pengembang, apapun itu pekerjaan yang dikerjakan oleh masing-masing anggota tim; tidak ada pengecualian untuk aturan yang satu ini;
- Individu Tim Pengembang boleh memiliki spesialisasi keahlian dan fokus di satu area tertentu, namun hasil dari

pekerjaan secara keseluruhan adalah milik Tim Pengembang.

- Tim Pengembang tidak memiliki sub-tim yang dikhususkan untuk bidang tertentu seperti pengujian atau analisa bisnis.

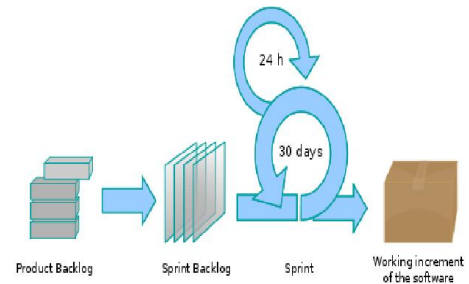
Pemilik Produk (*Product Owner*)

Pemilik Produk bertanggung jawab untuk memaksimalkan nilai dari produk dan hasil kerja dari Tim Pengembang. Cara untuk melakukan ini akan beragam di setiap organisasi, Tim Scrum, dan masing-masing individu.

Pemilik Produk adalah orang yang bertanggung-jawab untuk mengelola Product Backlog.

Pengelolaan Product Backlog mencakup:

1. Menjabarkan item Product Backlog secara jelas;
2. Mengurutkan item-item di dalam Product Backlog untuk dapat mencapai misi dan tujuan dengan cara terbaik;
3. Menilai hasil pekerjaan dari Tim Pengembang;
4. Memastikan bahwa Product Backlog kelihatan, transparan, dan jelas bagi semua pihak, dan menentukan Product Backlog mana yang harus dikerjakan selanjutnya oleh Tim Scrum;
5. Memastikan bahwa Tim Pengembang dapat memahami item di dalam Product Backlog hingga batasan yang mampu diberikan oleh Pemilik Produk.



Gambar Model Alur Scrum

1. Product Backlog

Product Backlog adalah daftar urutan dari semua yang perlu ada di dalam produk dan merupakan sumber utama dari daftar kebutuhan untuk semua perubahan yang perlu dilakukan terhadap produk. Pemilik Produk bertanggung-jawab terhadap Product Backlog, termasuk isinya, keberadaannya dan urutannya.

Product Backlog sifatnya tidak pernah habis. Pengembangan di awal-awal hanya menjabarkan daftar kebutuhan awal dan yang paling dipahami. Product Backlog berkembang seiring dengan berkembangnya produk dan lingkungan dimana ia berkembang. Product Backlog bersifat dinamis; senantiasa berubah untuk menentukan apa yang dibutuhkan oleh produk untuk dapat menjadi layak, kompetitif, dan berguna. Selama produk itu ada maka Product Backlog juga ada.

Product Backlog menjabarkan semua fitur, fungsi, kebutuhan, penyempurnaan dan perbaikan untuk perubahan yang akan dibuat terhadap produk di rilis mendatang. Item Product Backlog memiliki atribut deskripsi, urutan, dan estimasi.

Product Backlog diurutkan berdasarkan nilai, resiko, prioritas dan keterdesakan. Item urutan teratas dari Product Backlog mendapatkan perhatian paling utama dalam aktifitas pengembangan. Semakin besar pertimbangan, konsensus dan nilai terhadap Product Backlog tersebut maka semakin tinggi pula urutannya.

Item Product Backlog di urutan paling atas lebih jelas dan lebih rinci dibandingkan dengan item di urutan paling bawah. Semakin presisi estimasi yang dibuat berdasarkan informasi yang jelas dan rinci, semakin tinggi pula urutannya. Item Product Backlog yang akan dikerjakan oleh Tim Pengembang pada saat Sprint lebih jelas dan telah dibelah sedemikian rupa sehingga masing-masing item dapat di-“Selesai”-kan di dalam satu Sprint. Product Backlog yang telah dinyatakan

2. Sprint Backlog

Sprint Backlog adalah sekumpulan dari item Product Backlog yang telah dipilih untuk dimasukkan ke dalam Sprint dan rencana untuk menyelesaikan potongan produk dan merealisasikan tujuan Sprint. Sprint Backlog adalah prakiraan yang dibuat oleh Tim Pengembang mengenai fungsionalitas apa yang akan tersedia di potongan produk selanjutnya dan pekerjaan yang perlu dilakukan untuk menghasilkan fungsionalitas tersebut.

Sprint Backlog menjabarkan pekerjaan yang akan dilakukan oleh Tim Pengembang untuk

merubah item Product Backlog menjadi sebuah potongan produk yang “Selesai”. Sprint Backlog membuat semua pekerjaan yang dipilih oleh Tim Pengembang guna mencapai tujuan Sprint dapat dilihat.

Sprint Backlog adalah rencana dengan perincian yang cukup yang berubah seiring dengan bertambahnya pemahaman tim di Pertemuan Harian. Tim Pengembang memodifikasi Sprint Backlog sepanjang Sprint berlangsung dan Sprint Backlog pun berkembang sepanjang Sprint. Perubahan ini terjadi seiring dengan bekerjanya Tim Pengembang dan semakin meningkatnya wawasan tim untuk mencapai tujuan Sprint.

Dengan bertambahnya pekerjaan, tim Pengembang menambahkannya ke dalam Sprint Backlog. Seiring dengan dikerjakan atau selesainya pekerjaan, sisa pekerjaan yang telah diestimasi juga diperbaharui. Elemen dari perencanaan dikeluarkan ketika elemen tersebut dirasakan tidak dibutuhkan lagi. Hanya Tim Pengembang yang dapat merubah Sprint Backlog pada saat Sprint sedang berlangsung. Sprint Backlog sangat transparan, menggambarkan secara real-time pekerjaan yang akan diselesaikan oleh Tim Pengembang pada saat Sprint dan ia sepenuhnya menjadi milik Tim Pengembang.

3. Sprint

Detak jantung dari Scrum adalah Sprint, sebuah batasan waktu yang berjangka dari satu bulan atau kurang dimana potongan produk yang

“Selesai” dapat digunakan dan berpotensi untuk dirilis dibuat. Sprint memiliki durasi yang konsisten sepanjang masa waktu proyek atau pengembangan. Setiap Sprint langsung bergulir pada saat Sprint yang sebelumnya telah selesai.

Sprint memiliki dan terdiri dari Pertemuan Perencanaan Sprint, Pertemuan Harian, pengembangan, Pertemuan Ulasan Sprint, dan Pertemuan Refleksi Sprint.

Pada saat Sprint sedang berlangsung:

- Tidak boleh ada perubahan yang berimplikasi terhadap tujuan Sprint.
- Komposisi dari Tim Pengembang dan kualitas dari tujuan harus senantiasa konstan.
- Ruang lingkup dapat diklarifikasi dan dinegosiasi ulang antara Pemilik Produk dan Tim Pengembang seiring dengan semakin banyaknya pengetahuan yang telah didapatkan sepanjang pengembangan.

Setiap Sprint dapat dikatakan merupakan sebuah proyek yang memiliki jangka waktu tidak lebih dari sebulan. Sebagaimana halnya proyek, Sprint juga digunakan untuk sebuah pencapaian. Setiap Sprint memiliki definisi mengenai apa yang akan dikembangkan, sebuah disain dan rencana luwes yang akan memandu bagaimana membangun produk, pekerjaan, dan hasil produk.

Sprint dibatasi hingga satu bulan kalender. Ketika

jangka waktu dari Sprint terlalu lama maka definisi mengenai apa yang sedang dikembangkan dapat berubah, kompleksitas akan bertambah, dan resiko juga mungkin akan bertambah. Sprint memungkinkan peristiwa menjadi lebih dapat diprediksi dengan memastikan pelaksanaan peninjauan dan penyesuaian kemajuan menuju tujuan setidaknya setiap satu bulan sekali. Sprint juga membatasi resiko biaya cuma hingga satu bulan saja. (Ken Schwaber and Jeff Sutherland, 2013)

Untuk memudahkan monitoring, setiap tim membuat taskboard yang berisi user story, todo, in process, to verify, dan done. Berikut contoh tasboard.

Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... 9 Test the... SC 4 Test the... SC 6 Test the... SC 6
	Code the... 2	Code the... 8	Test the... SC 8		
	Test the... 8	Test the... 4			
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC 8 Test the... SC 6 Test the... SC 6
	Code the... 4	Code the... 6			

Gambar Taskboard

Scrum tepat digunakan saat kondisi:

- Keperluan berubah dengan cepat
- Tim programmer sedikit, yaitu 5-9 orang
- Pelanggan tidak terlalu paham dengan apa yang diinginkan

Kelebihan Scrum antara lain:

- Keperluan berubah dengan cepat
- Tim berukuran kecil sehingga melancarkan

komunikasi, mengurangi biaya dan memberdayakan satu sama lain

- Pekerjaan terbagi-bagi sehingga dapat diselesaikan dengan cepat
- Dokumentasi dan pengujian terus menerus dilakukan setelah software dibangun
- Proses Scrum mampu menyatakan bahwa produk selesai kapanpun diperlukan

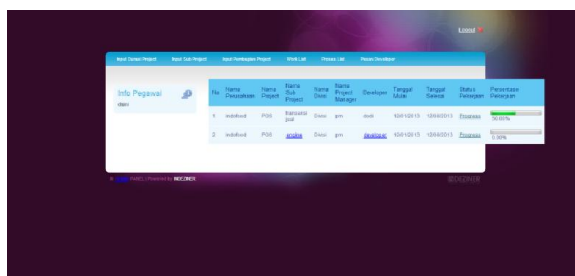
Kelemahan Scrum antara lain:

1. Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima.

IV PEMBAHASAN DAN IMPLEMENTASI

4.1 Desain Proses list

							Logout
Input Sub Project	Input Durasi Project	Input Pembagian Project	Worklist	Proses List	Pesan		
Info Pegawai	Nama Perusahaan XXXX	Nama Project XXXX	Nama Sub Project XXXX	Nama Project Manager XXXX	Tgl Mulai 09-09-9999	Tgl Selesai 09-09-9999	Progress



Halaman ini digunakan untuk melihat sampai mana progress pengerjaan proyek yang sudah dilakukan oleh developer

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasar penelitian yang telah dilakukan maka dapat disimpulkan sebagai berikut :

1. Aplikasi Project Management pada Blung Studio ini akan membantu dalam pengerjaan project yang ada pada Blung Studio agar project dapat terpantau perkembangannya dan dalam pengerjaan project bisa selesai sesuai dengan estimasi waktu yang telah ditentukan.
2. Dengan adanya Aplikasi Project Management pada Blung Studio ini diharapkan mampu memberikan informasi kepada developer atau karyawan project apa yang akan dikerjakan dan waktu pengerjaannya.

5.2 Saran

Saran yang menjadi masukan untuk meningkatkan kinerja aplikasi Project Management pada Blung Studio adalah :

Aplikasi ini bisa dikembangkan lebih lanjut dengan menggunakan metode dan bahasa script atau bahasa pemrograman yang lain sehingga dalam mengelola project dapat dihasilkan informasi yang sesuai dengan kebutuhan yang akan datang pada Blung Studio.

DAFTAR PUSTAKA

- [1]Widjaya,Iwan Kurniawan,S.Kom.,M.Kom.,M.T., (2013).*Manajemen Proyek Teknologi Informasi*.Yogyakarta:Graha Ilmu.
- [2]Santosa,Budi(2009).*MANAJEMEN PROYEK: Konsep & Implementasi*. Yogyakarta:Graha Ilmu.
- [3] http://id.wikipedia.org/wiki/Server_web (diakses tanggal 5 April 2013)
- [4]Nugroho,Bunafit(2009).*Membuat Aplikasi Web PHP dan MySQL*. Yogyakarta:Gava Media.

- [5] Hanif Al Fatta. (2009). *Analisis dan Perancangan Sistem Informasi berbasis OOP*. Yogyakarta: Penerbit Andi
- [6] http://id.wikipedia.org/wiki/Unified_Modeling_Language, di akses tanggal 21 September 2013
- [7] <http://ifnubima.org/buku-scrum-dan-xp-secara-praktis/>, diupdate tanggal 1 Mei 2013
- [8] Schwabber, Ken Jeff
Sutherland., (2011). *Scrum Guide ID*. Terjemahan: Joshua Partogi (2013). Jakarta